Anisotropic propagation of user interests in ontology-based user models

Anisotropic propagation of user interests in ontology-based user models

Federica Cena, Silvia Likavec, Francesco Osborne

Dipartimento di Informatica, Università di Torino, Italy
Corso Svizzera 185, 10149 Torino, Italy
{cena,likavec,osborne}@di.unito.it

Abstract

This work contributes to the development of ontology-based user models, devised as overlays over conceptual hierarchies derived from domain ontologies. We tackle the problem of propagation of user interests in such a conceptual hierarchy. In addition to accounting for the hierarchical structure of the domain and the type and amount of feedback provided by the user, the principal contributions introduced in this work are: (i) *horizontal* propagation which enables propagation among siblings, in addition to vertical propagation among ancestors and descendants; (ii) *anisotropic* vertical propagation which permits user interests to be propagated differently upward and downward; (iii) *context-dependance* which introduces the possibility to propagate differently according to various contexts for specific applications; (iv) *support for dynamic ontology maintenance*, i.e. preserving the user interest values when adding or removing a node from the conceptual hierarchy. Our approach supports finer recommendation modalities and contributes to the resolution of the cold start problem, since it allows for propagation from a small number of initial concepts to other related domain concepts by exploiting the conceptual hierarchy of the domain. A field evaluation confirmed the effectiveness of our approach w.r.t. the traditional vertical propagation.

Keywords: user model, ontology, conceptual hierarchy, propagation of interests

1. Introduction

With the advent of the Semantic Web, ontologies emerged as powerful means for representing domain knowledge in many areas [6, 4], such as cultural heritage, tourism, e-commerce, e-learning, digital libraries, web search etc. One of the promising uses of ontologies is in the area of adaptive systems, such as recommender systems and adaptive search engines, in which the final services are tailored to users' needs to account for differences in their individual requirements. These requirements are obtained from users' features, contained in the User Model [26]. In particular, the user model should maintain the user's attitude (such as interest, knowledge) in the main domain objects. It has been devised as particularly useful to represent the domain objects as classes of an ontology and the user model as overlay over such an ontology, giving rise to

ontology-based user models. This is particularly useful since the conceptual hierarchy derived from domain ontology can be exploited for different forms of interest propagation (from ancestors to descendants and vice versa and among siblings) used further to infer missing values for user interest in some ontological concepts. The work described in this paper contributes to the development of ontology-based user models and deals with the propagation of user interests in the ontology. In particular, we consider a specific type of ontology, namely conceptual hierarchy derived from domain ontology.

We build upon our previous work presented in [14] where we introduced OBUM - Ontology Based User Model, and an approach for propagating interests in a user model designed as an overlay over the domain ontology constructed as a conceptual hierarchy. In this previous approach, the conceptual hierarchy of the domain and the type and amount of feedback provided by the user were taken into account in order to propagate user interest among domain concepts, starting from a few feedback values for some of them. In the present work, in order to make the approach more effective and overcome some of its previous limitations, we introduced various improvements for the propagation of user interests in the conceptual hierarchy:

- *horizontal* propagation which enables propagation among siblings, in addition to propagation among ancestors and descendants;
- *anisotropic* vertical propagation which permits user interests to be propagated differently upward and downward;
- *context-dependance* which introduces the possibility to change the propagation according to various contexts for specific applications;
- *support for dynamic ontology maintenance*, i.e. the possibility to preserve the user interest values when adding or removing a node from the conceptual hierarchy, hence making it possible to apply our approach not only to static ontologies but also to the ontologies that change dynamically;
- new field evaluation that confirms the effectiveness of our approach.

In the rest of the paper, we give a detailed account of all these aspects, organizing the paper as follows. First, in Section 2, we provide the background notions employed when developing our approach, together with the most important state of the art work in the corresponding fields. Next, we provide the details of our approach for user interest propagation in Section 3, explaining the sensed and propagated interest values and vertical and horizontal propagation. The algorithms are explained in Section 4, together with an analysis of their correctness, termination, stability. In Section 5 we illustrate the interest values calculation for a particular use case. In Section 6 we describe our approach to possible ontology maintenance. The results of a real field evaluation can be found in Section 7, followed by the comparative analysis with related work in the field in Section 8. We conclude in Section 9 giving some directions for possible future development.

2. Background and state of the art

In this section we provide a brief account of the background notions used to develop our approach. For each related field, we indicate the most relevant work in the literature, providing the state of the art we started from.

2.1. Ontologies and conceptual hierarchies

Ontologies are powerful formalisms for knowledge representation which enable explicit specifications of domain concepts with their properties and relationships between them. In theory, an ontology can be seen as a "formal, explicit specification of a shared conceptualization" [20]. Ontologies provide exact semantics for each statement, avoiding any semantic ambiguities. Moreover, being expressed with standard formats and technologies, they allow for extensibility and re-usability. Finally, the associated rigorous reasoning mechanisms allow for different forms of reasoning (for example, to deduce implicit classes).

A simplified view of an ontology is a network of concepts. It provides a natural basis for modeling a domain of discourse following multiple examples in the area of AI and Intelligent Systems. In our work, we focus on a particular type of ontology, namely *conceptual hierarchy* derived from the domain ontology, also known as *hierarchical ontology*. This kind of ontology is a taxonomy of concepts where concepts are organized based on the partial order relation IS-A, through which entities are grouped into or subsumed by a higher level classes [9, 43]. A conceptual hierarchy can be seen as a simple ontology where the properties of concepts are not taken into account. In this sense the term "ontology" includes the term "conceptual hierarchy". Hierarchical ontologies are important since they enable structuring of the domain knowledge into categories and formulation of relations among concepts in an abstract way which enables easier development, maintenance and reuse. Also, inference mechanisms and reasoning are faster and more efficient.

2.2. User Model

Adaptive web systems belong to the class of user-adaptive software systems, which provide each user with personalized services, i.e. they behave differently for different users [26, 1, 11]. For example, when the user searches for information, the system can adaptively select the most relevant items [29]. When the user navigates a web site, the adaptive system can manipulate the links (e.g. hiding, sorting, annotating a link) to provide adaptive navigation support [10]. When the user reaches a particular page, the system can present the content adaptively [12].

Adaptive results are made possible due to the usage of a User Model (UM), a representation of information about an individual user. The process of building and maintaining a user model is known as "user modeling" process [11]. To create and maintain an up-to-date user model, an adaptive system collects data for the user model from various sources which may include implicitly observing user interaction [26], exchanging user data with other applications [7], using user information provided by social networking sites via APIs [38] or/and explicitly requesting direct input from the user.

Another step in the user modeling process is finding the best way to represent the content in the user model: using simple vectors or bags of words [1], property-value pairs [13], complex probabilistic approaches (such as bayesian networks [2]), or overlay over the domain model [11], where the user's current state (e.g. interest or knowledge) with respect to domain concepts is recorded.

2.3. Ontologies for user modeling

Ontologies (and conceptual hierarchies) started to be widely used in adaptive systems to represent the domain models [41, 31]. Their usage allows the internal knowledge of adaptive systems to be sharable [33, 5], and offers a great opportunity to improve the user modeling process.

Ontologies have been exploited to represent directly the user model features (demographic features such as age, gender, profession, etc.) in a taxonomical structure. Examples of this approach are the General User Model Ontology (GUMO) [22], the Unified User Context Model (UUCM) [28], the User Role Model ontology [45] and OntobUM [33]. This allows for simpler interoperability among adaptive systems which want to exchange user model data.

When a domain model is represented as an ontology, the user model can be built as an overlay over the ontology [41, 21, 24]. This ontology-based overlay user model consists of a subset of concepts from the domain ontology, where the presence of a concept in the user model means that the user has an interest in the corresponding concept. The concepts can have associated values (weights, verbal labels), characterizing how much the user is interested in them. The overlay model of user interests can be represented as a set of pairs $\langle concept, value \rangle$, with one pair for each domain class (for example, $\langle wine, 0.9 \rangle$ on a scale $0 - 1$ means strong interest in wine).

2.4. Motivations for our work

Ontology-based overlay user model is a great opportunity for user modeling, since matching the user interests with the concepts in the domain ontology allows to benefit from the conceptual structure of the ontology and attain the values for other concepts in the ontology [30]. For example, starting from user interest in a class, we can propagate this value to related concepts (subclasses, superclasses and concepts at the same level). More specifically, such an ontological approach to user profiling brings the following benefits:

- *Solution of the cold start and sparsity problem*. The *cold start* problem [37] happens at the beginning of the interaction when the system does not have enough user data to provide appropriate adaptation. *Sparsity* problem is a related issue that occurs when the available data are insufficient for identifying user interest and it is a major issue that limits the quality of recommendations and the applicability of adaptation in general. With this approach, it is possible to derive many missing values starting from a few initial user interest values, and thus solving both problems.
- *Adaptation accuracy improvement*. This approach is useful not only in the early phases of interaction, but also in all the subsequent phases of the adaptation process. In fact, propagation allows to have more information about users making it possible to recommend to users the objects which might interest them more or to personalize search results.

In this work, we propose to exploit the opportunity to devise a user model as an overlay over a conceptual hierarchy for solving the *cold start problem* and to *improve adaptation results* in adaptive systems. The details of our approach will be given in the following section.

3. User interest propagation

In this paper we present an approach for propagation of user interests in a conceptual hierarchy derived from domain ontology (referred to as conceptual hierarchy from now on) in order to obtain missing information about the user (i.e the interest in domain concepts), and thus enrich the user model. The approach uses the notions described in the previous section, and in particular:

- the domain model has to be represented using a conceptual hierarchy, modeling domain items as instances of the classes in the ontology;
- the user model has to be represented as an overlay over domain model represented as a conceptual hierarchy: in particular, we propose to represent the user interest as a weight which indicates how much the user is interested in each concept.
- the system must have a way to obtain user feedback about interests and preferences of users, either explicitly or implicitly.

The procedure for user interest propagation consists of the following basic steps:

1. Calculate the *conceptual distance* between each pair of concepts in the conceptual hierarchy (Section 3.1).
2. Capture the *user feedback* for some concepts in the hierarchy (Section 3.2).
3. Determine the *sensed interest value* as a measure of how much of the direct user feedback each node can *sense*. Sensed interest depends on the total amount of direct user feedback for the given node and the level of the node in the hierarchy (Section 3.3)
4. Calculate the *propagated interest value* and propagate it vertically and horizontally in the conceptual hierarchy. Propagated interest depends on the sensed interest of the starting node (i.e. the node receiving the feedback) and on the conceptual distance between the starting node and the node receiving the propagated value (Section 3.4).
5. Update the *total interest value* for each node in the hierarchical ontology. Total interest value is the sum of the sensed interest value and all the propagated interest values received (Section 3.5).
6. For each subsequent user feedback, calculate the difference between the newly obtained sensed interest value for the given node and the old one, so as to propagate only the appropriate portion of the sensed interest value (Section 4).

Hence, after collecting the feedback provided by the users, each node that received direct feedback from a user re-calculates its own *sensed interest*. The change in the previous sensed interest value will be propagated *vertically* and *horizontally* to the neighboring nodes and update the user model. Each node which receives the propagated interest will update its *total interest value*.

Our propagation of user interests can be seen as a special case of spreading activation theory [36]. The cornerstones of spreading activation are: (i) determining the initial set of nodes and their corresponding activation values, (ii) establishing the weights of the relations in the network, (iii) establishing the firing threshold and decay factors. In our approach the initial set of nodes is the set consisting of the nodes which received the user feedback, taking their sensed interest values as their initial activation values. Our conceptual hierarchy is an associative network in which each edge is assigned a

numeric weight corresponding to the conceptual distance between the two nodes. We do not account for the firing threshold (any initial value is propagated, but only from the initial node). Also, in order to account for the decay factor in spreading activation we simulate attenuation law in physics. This means that from the initial set of nodes which received the user feedback as the signal of user interest, adequately diminished interest values would be propagated to the related nodes in the hierarchy. Another important issue is a termination condition for spreading activation. In our case, since only the nodes which received the feedback are propagating the interest values, and propagate them to a finite number of nodes, there is no danger of encountering loops.

In the rest of the section, we will describe all the steps of the propagation process and the algorithms, together with an analysis of their correctness, termination, stability. Finally, we will present implementation details of our propagation algorithm.

3.1. Conceptual distance

The first step regards the calculation of distance among concepts. In a conceptual hierarchy, the domain objects at the lower levels are closer than those with the same relationship located at higher levels. For example, the wine *Barbera d'Alba* and its particular variety *Pio Cesare "Fides" Barbera d'Alba 2000* are closer than *Barbera d'Alba* and *Red Wine*, concepts higher up in the conceptual hierarchy. Thus, it does make sense to *propagate user interests differently in the conceptual hierarchy*, particularly when propagating upwards and downwards or to siblings, which brings us to *anisotropic propagation* of user interests.

To achieve this, it is necessary to differentiate among concepts according to some conceptual similarity measure. In the previous version of our work [14], we considered uniform distances in the conceptual hierarchy, calculating the distances between two concepts by counting the nodes or the edges between the two concepts. This distance was then used to determine the similarity between concepts [32]. Now, instead of considering all distances equal, we modified our approach by using the *conceptual distance* and *conceptual similarity* introduced in [16] which provided us with the immediate solutions to the above mentioned considerations.

We give here a brief summary of the approach for computing conceptual distances among domain objects [16]. Ontology is considered a rooted, directed acyclic graph of subclass-of and instance-of relations. A new notion of *specificity* (SPEC) is introduced to be able to distinguish the concepts relevant in a certain context. The specificity of the irrelevant concepts is equal to infinity. *Surface nodes* are the first domain-relevant concepts encountered when traversing the domain ontology with a finite specificity $\text{SPEC}(n) \in \mathbb{N}$. If a node n is reachable from a single surface node s via a single path p, $\text{SPEC}(n)$ is obtained by adding the number of edges in p to $\text{SPEC}(s)$ (and if there are more paths of different lengths, the longest path is used).

Specificity is used to calculate *conceptual distances* between objects of the ontology, introducing exponentially decreasing edge lengths, as opposed to all the edges of the same length. In particular, given an edge $e : S \rightarrow T$, the edge length is given by

$$\text{LEN}(e) = c^{-\text{SPEC}(S)}$$

where $c \in \mathbb{N}$ is a constant ($c \geq 2$). The length of a path is the sum of the lengths of its edges. Then, given two ontology nodes N_1, N_2, we calculate their conceptual distance

$\overline{dist}$ as the length of the shortest path[1] connecting them via an ancestor node:

$$\overline{dist}(N_1, N_2) = \min\{\text{LEN}(p_1) + \text{LEN}(p_2) \mid \exists G \text{ such that } G \rightarrow_{p_1} N_1, G \rightarrow_{p_2} N_2\}.$$

Whenever a path from N_1 to N_2 crosses a non-relevant node, then $\overline{dist}(N_1, N_2) = \infty$.

The choice of surface nodes depends on the context of the application and the level of detail one wants to include in the given situation. To enable this *context dependence*, in [16] it is possible to choose the surface nodes according to the application context: for example, *Wine* could be given a lower specificity in the context of a wine festival, while in any other non wine-specific context it could be given the default value 0. For a detailed account of specificity and conceptual distance we refer the reader to [16].

3.2. Capturing user feedback

Capturing user feedback is very important, but in this work we do not go into details of how user feedback is obtained. In many web-based adaptive systems, it is possible to infer user interests indirectly by observing user interaction with the web-based system, without explicitly asking the user for information. In our approach, we consider user actions (selecting, tagging, bookmarking, commenting an item, etc) as indicators of user interest [26] and we assign to each of these actions a certain weight f expressing how much it indicates the strength of user interest (following [13]).

3.3. Sensed interest value

When a user performs an action on a certain domain object, the node corresponding to that object "senses" a certain amount of the provided feedback. This value is what we call a *sensed interest*, thus introducing the concept of a *sensitivity* of a node, i.e. how much a node can react to user feedback. The sensitivity of a node depends on its position in the conceptual hierarchy and user feedback for that node. The former is important since the hierarchical structure of the ontology represents different levels of generalization [34]. Thus, when a user expresses interest in an object lower down in the conceptual hierarchy, it should have a stronger impact on the user model than interest expressed in more generic concepts, since it provides the system with more precise information. Following this direction, in our approach a lower amount of feedback is required as a signal of strong user interest for a lower-level node.

In order to calculate the interest sensed by a node N, we use the following formula:

$$\mathcal{I}_{\mathcal{S}}(N) = \frac{\ell(N)}{max} sig(F(N) + f(N)) \tag{1}$$

where

- $\ell(N)$ is the level of the node that receives the feedback,
- max is the level of the deepest node in the conceptual hierarchy,
- $sig(x)$ is a variant of the sigmoid function given by $sig(x) = \dfrac{2}{1 + e^{-x}} - 1 = \dfrac{e^x - 1}{e^x + 1}$,

[1] considering the graph undirected.

- $F(N)$ is the sum of the previous feedback values obtained from the user for the given node N,
- $f(N)$ is the current (new) feedback obtained from the user for the node N.

We decided to use the sigmoid function since it permits to take into account also negative user feedback. In this paper, we do not consider negative user feedback, but we wanted to propose the approach as general as possible. Using this sigmoid function, the above Formula 1 can be written as:

$$\mathcal{I}_{\mathcal{S}}(N) = \frac{\ell(N)}{max} \frac{e^{F(N)+f(N)} - 1}{e^{F(N)+f(N)} + 1}. \tag{2}$$

Note that we calculate the level of the node in the conceptual hierarchy, as well as the level of its deepest node, starting from the *surface nodes*, while the rest of the nodes is not taken into account (see Section 3.1). This means that the surface nodes are considered to be at level zero. As a consequence, the surface nodes do not sense anything, since the sensed interest for these nodes is always equal to zero. This also means that the surface nodes are not able to propagate anything. This is intuitively correct, since the surface nodes are the least significant concepts in the given context. This brings us back to the importance of context dependence, i.e. the possibility to chose the surface nodes based on the context of application usage.

The order in which the user feedback is received does not influence the sensed interest value. This is an important point corrected with respect to the previous approach [14]. In fact the order of the actions performed by the user should not have any implications on the interest values of the user. For the same reason, the node does not distinguish if it receives one feedback f or many single ones $f_1, \ldots, f_n$ such that $f_1 + \cdots + f_n = f$.

3.4. Propagated interest value

We first explain the concepts of *vertical* and *horizontal propagation*, followed by the description of how to calculate the propagated interest values.

Vertical propagation. In the case of vertical propagation, user interest in a certain domain concept can be propagated upward to its antecedents, as well as downward to its descendants. For example, if a person is interested in red wine *Barbera_d'Alba*, we can assume that the same person might be interested in a specific kind of this wine, such as *Pio_Cesare_Fides_Barbera_ d'Alba_2000*, but also in *Red_Wine* in general. The interest for the neighboring concepts is smaller than the interest in the original concept that received the feedback and it diminishes with increasing distance from the initial node. Hence, we can assume that the interest we can propagate vertically is inversely correlated to the distance from the node that received the feedback. Another effect we want to obtain is to have *anisotropic propagation*, i.e. stronger propagation downwards than upwards in the hierarchy. This is allowed by using the exponentially decreasing distance described in Section 3.1.

Horizontal propagation. It is also useful to be able to propagate horizontally, propagating the user interests also to the sibling nodes of the node that receives the feedback. For example, if a person is interested in red wine *Barbera_d'Alba*, the same person

could probably be also interested in its sibling *Barbera_d'Asti*. Again, we are less sure about this interest and thus this value will be smaller.

Since both vertical and horizontal propagation are computed only by using the shorter distance between the node that receives the feedback and the node that receives the propagated interest value the procedure for calculating the propagated interest is the same in both cases. The propagation of interests follows the conventional law which rules the natural phenomena, where the attenuation is proportional to the initial value of the variable and to the taken distance in a medium (see for instance DeBeer-Lambert law [42] for light attenuation). The intensity is not a linear function of the taken distance, rather an exponential one. This formulation allows to obtain a decrease of the interest value with respect to the distance from the initial node. Hence, for the initial user feedback, we compute the initial propagated interest $\mathcal{I}_{\mathcal{P}}(M, N)$ from the node M to the node N as follows:

$$\mathcal{I}^0_{\mathcal{P}}(M, N) = \mathcal{I}_{\mathcal{S}}(M)e^{-k \cdot dist(M,N)} \tag{3}$$

where

- $\mathcal{I}_{\mathcal{S}}(M)$ is the sensed interest of the node M that received the feedback,
- $dist(M, N)$ is the *conceptual* distance (i.e. exponentially decreasing distance) between the node M receiving the feedback and the node N receiving the propagated interest,
- $k \in \mathbb{R}$ is a constant, called *attenuation coefficient*.[2]

Only with the initial user feedback for a certain domain object, all the sensed interest is used in propagation. With each subsequent user feedback, the sensed interest used to update the user model is equal to the difference between the newly obtained sensed interest value for the given object and the old one that was already propagated:

$$\Delta\mathcal{I}_{\mathcal{S}}(N) = \mathcal{I}^t_{\mathcal{S}}(N) - \mathcal{I}^{t-1}_{\mathcal{S}}(N) = \frac{\ell(N)}{max}\Big(sig(F(N) + f(N)) - sig(F(N))\Big) \tag{4}$$

where t and $t-1$ are two successive moments when the user feedback was received.

It is easy to verify that the amount of interest $\Delta\mathcal{I}_{\mathcal{S}}(N)$ propagated for the i-th feedback is lower than the amount of interest propagated for any previous feedback with the same values. This helps to keep the system balanced avoiding the propagation of redundant information. Therefore, for each subsequent feedback for the node M, the propagated interest $\Delta\mathcal{I}_{\mathcal{P}}(M, N)$ to the node N is calculated as:

$$\Delta\mathcal{I}_{\mathcal{P}}(M, N) = e^{-kdist(M,N)}\Delta\mathcal{I}_{\mathcal{S}}(M). \tag{5}$$

Since for the initial feedback, there is no previous feedback, we obtain $\Delta\mathcal{I}_{\mathcal{S}}(N) = \mathcal{I}_{\mathcal{S}}(N)$, and $\mathcal{I}^0_{\mathcal{P}}(M, N) = \Delta\mathcal{I}_{\mathcal{P}}(M, N)$. Hence, we can always use the value $\Delta\mathcal{I}_{\mathcal{P}}(M, N)$ when updating the user model with propagated interest.

[2] In physics, k depends on the characteristics of the medium relevant for the process, for instance opacity of the material for a light wave.

In order to correctly propagate user interests, the contribution of the various nodes to the propagation must be balanced. In particular, the node contribution to the propagation process should decrease with the amount of feedback already propagated by that node. This is needed to prevent the non-proportional propagation of interests when the user concentrates solely on one or a few concepts. In that case, this input is redundant and does not add any information. In our approach this is avoided since the total sensed interest $\mathcal{I}_{\mathcal{S}}(N)$ is a sigmoid function which asymptotically saturates at $\pm\frac{\ell(N)}{max}$. This makes it impossible for a single node to unbalance the user model and it also rewards those cases in which a node receives feedback from a good part of its sub-nodes, showing a consistent interest for that class.

In the previous version of our approach [14], the propagated interest value was inversely correlated with the amount of feedback already received by the node. It was the first try to avoid unbalancing the user model by redundant feedback. However, this approach had some flaws. It depended on the number of user actions and not on the total amount of feedback, unfairly penalizing the items which received many little positive feedback values. In addition, the total interest in a domain object depended on the order in which the individual feedbacks were provided. As stated before, we solved the problem by making $\mathcal{I}_{\mathcal{S}}(N)$ a sigmoid function.

One dilemma is whether to propagate user feedback between two nodes which only have a surface node as their common ancestor. For example, when a user feedback for *White_Wine* is propagated to *Red_Wine* through the common ancestor surface node *Wine*. Depending on how the surface nodes are chosen, this might not be desirable in certain scenarios. To avoid this problem it is possible to set a *maximum propagation distance maxP* equal to the distance of two nodes which have a surface node as their only ancestor. Since the distances decrease exponentially with the depth of the ontology, this limit does not preclude the propagation between any other pair of nodes.

3.5. Total interest value

Finally, for each node N we calculate its total interest value, taking into account its sensed interest value and the propagated interest values received from other nodes:

$$\mathcal{I}(N) = \sigma\mathcal{I}_{\mathcal{S}}(N) + \Sigma_{i=1}^{n}\pi_i\mathcal{I}_{\mathcal{P}}(N_i, N) \tag{6}$$

where $\mathcal{I}_{\mathcal{S}}(N)$ is the sensed interest for the node N and $\mathcal{I}_{\mathcal{P}}(N_i, N)$ is the total propagated interest from the node N_i to the node N, n being the number of nodes which propagate their interest values to the node N. $\sigma, \pi_i \in \mathbb{R}$ are constants such that $\sigma + \Sigma_{i=1}^{n}\pi_i = 1$ which permit to assign different importance to sensed and propagated interest values.

4. The propagation algorithm

In this section we look into more details of the propagation algorithm and its performance, as well as its correctness, termination and stability. Note that we are always working only on the relevant part of the conceptual hierarchy, i.e. we only consider the surface nodes and their descendants.

The algorithm consists of two distinct procedures. The first one is *distance matrix computation* (DMC), described by Algorithm 1, which takes as input the ontology describing the domain and returns a matrix DM which associates to any pair of nodes the conceptual distance between them (see Section 3, step 1). In the first step, it uses the ontology reasoner to infer the hierarchical structure and make explicit the subsumption relationships between classes (line 2). It then exports this conceptual hierarchy to a graph in which nodes are the classes of the ontology and edges are the IS-A relations (line 3). Each graph edge is then assigned an exponentially decreasing weight as described in Section 3.1 (line 4). Finally the procedure exploits the well-known three nested for-loops formulation of the Floyd-Warshall algorithm [23] to find the shortest paths between each pair of nodes and create the conceptual distances matrix DM (line 5). The matrix is saved in a binary file to be used by the interest propagation algorithm. This procedure runs on the conceptual hierarchy at the beginning and should be run again every time the ontology is modified, as described in Section 6.

Algorithm 1 Distance matrix computation

```
1: function DMC (ontology) returns (DM)
2: inferHierarchicalStructure(ontology);
3: graph = extractTaxonomy(ontology);
4: graph = assignWeights(graph);
5: DM = FloydWarshall(graph);
6: return DM;
```

The second procedure is the *interest propagation*, shown in Algorithm 2, which uses the pre-computed distance matrix DM, user feedback values and users to update the user models of the users providing the feedback (see Section 3, steps 2-6). The interest propagation can be run once a day, but, if computational power is not an issue, it may be run at shorter intervals and even after every user feedback.

The algorithm assigns the value zero to all the received propagated interest values for all users. It then loops through all the feedback values and for each of them selects the node N in the conceptual hierarchy on which the action was performed (line 8) and determines the user U which provided the feedback (line 9). For the node which received the feedback, the sensed interest value is updated using the Formula 1 (line 10) and the total interest value is calculated as a weighted sum of sensed interest and the received propagated interest (line 11). For all the other nodes in the conceptual hierarchy i.e. for all the nodes M in graph different from N the received propagated interest value and total interest value are updated with deltaIP(N,M) calculated according to Formula 5 (lines 13 and 14). Note that the constants a, b, c, d, e, f in Algorithm 2 have to be chosen so that $a + b = 1$, $c + d = 1$ and $e + f = 1$.

4.1. Implementation details

We implemented the described algorithm as a Java class which can be controlled via servlet or scheduled to run at chosen times on a 2.4 GHz Intel Core Due with 2 GB of

Algorithm 2 Interest Propagation

```
1: function interestPropagation (feedbacks, users)
2: for all N in graph do
3:    for all U in users do
4:       IP(U,N) = 0;
5:    end for
6: end for
7: for all (f in feedbacks) do
8:    N = nodeID(f);
9:    U = userID(f);
10:   IS(U,N) = l(N)/max sig(F(N)+f(N));
11:   I(U,N)=a IS(U,N) + b IP(U,N);
12:   for all M in (graph - N) do
13:      IP(U,M)=c IP(U,M)+d deltaIP(U,N,M);
14:      I(U,M)=e I(U,M)+f deltaIP(U,N,M);
15:   end for
16: end for
```

RAM. The domain ontology is written in OWL[3] and developed with Protege editor[4]. We used the Java library *Jena*[5] to parse the OWL Ontology and Pellet reasoner[6] to infer its hierarchical structure. We use our own lightweight implementation of a graph object, which gives access to the taxonomy faster than Jena OntModel. The user model values are stored in a MySQL database, hence the function updateUM directly updates a row in the table representing the user model. The same table is queried to obtain the user preferences for various concepts.

4.2. Algorithm correctness, termination and stability

Algorithm 1 uses the inferHierarchicalStructure procedure from Pellet reasoner and a well-known Floyd-Warshall algorithm, hence its correctness and termination is straightforward. We will only look into Algorithm 2.

Correctness of Algorithm 2

Claim: Each concept N in the user model of user U has the total interest value:

$$\mathcal{I}(N) = \sigma \mathcal{I}_{\mathcal{S}}(N) + \Sigma_{i=1}^{n} \pi_i \mathcal{I}_{\mathcal{P}}(N_i, N).$$

Proof: By induction on the number of feedbacks. Since we are proving the claim for a certain user U, and thus considering only the feedback values for this particular user, we would omit the argument U from algorithm's notation.

[3] http://www.w3.org/TR/owl-features/
[4] http://protege.stanford.edu
[5] *http://www.jena.apache.org/*
[6] *http://www.clarkparsia.com/pellet/*

- $k = 1$. We distinguish two cases:
  - Feedback f_1 is provided for the node N. Then according to Algorithm 2, its sensed interest is calculated using the value f_1 (line 10) and $I(N) = aIS(N) + bIP(N) = aIS(N)$ since $IP(N) = 0$ initially (line 4). So in this case $\sigma = a$ and $\pi_i = 0, i = 1, \ldots, n$ and

$$\mathcal{I}(N) = \sigma \mathcal{I}_{\mathcal{S}}(N).$$

  - Feedback f_1 is provided for some other node M. Then $I(N) = eI(N) + fdeltaIP(M, N) = fdeltaIP(M, N)$ since $I(N) = 0$ (this is the first feedback and it is given to another node). Also $deltaIP(M, N) = IP(M, N)$. In this case $\sigma = 0, \pi_1 = f$ and

$$\mathcal{I}(N) = \pi_1 \mathcal{I}_{\mathcal{P}}(M, N).$$

- Suppose the induction hypothesis holds for k. This means that

$$\mathcal{I}(N) = \sigma \mathcal{I}_{\mathcal{S}}(N) + \Sigma_{i=1}^{l} \pi_i \mathcal{I}_{\mathcal{P}}(N_i, N) \text{ i.e.}$$

$$I(N) = sIS(N) + r_1 IP(M1, N) + ... + r_l IP(M_l, N).$$

  where l is the number of other nodes which received the feedback and $IP(M_i, N)$ is the total received propagated value by the node N from the node $M_i, i = 1, \ldots, l$.
- We prove the induction hypothesis for $k + 1$. We distinguish two cases:
  - Feedback f_{k+1} is provided for the node N. Then according to Algorithm 2, its sensed interest is updated using the previous feedback values and value f_{k+1} (line 10) and $I(N) = aIS(N) + bIP(N) = aIS(N) + b(r_1 IP(M1, N) + ... + r_l IP(M_l, N))$. In this case $\sigma = a$ and $p_i = b \cdot r_i, i = 1, \ldots, l$ and

$$\mathcal{I}(N) = \sigma \mathcal{I}_{\mathcal{S}}(N) + \Sigma_{i=1}^{l} \pi_i \mathcal{I}_{\mathcal{P}}(N_i, N).$$

    Note that the number of nodes which propagated their interest to the node N remains l, since the feedback f_{k+1} was provided for the node N. Also, the value of the sensed interest for the node N was updated.
  - Feedback f_{k+1} is provided for some other node Q. Then $I(N) = eI(N) + fdeltaIP(Q, N) = e(sIS(N) + r_1 IP(M1, N) + ... + r_l IP(M_l, N)) + fdeltaIP(Q, N)$. So $\sigma = es, \pi_i = er_i, i = 1, \ldots, l$ and $\pi_{l+1} = f$. So we obtain

$$\mathcal{I}(N) = \sigma \mathcal{I}_{\mathcal{S}}(N) + \Sigma_{i=1}^{l} \pi_i \mathcal{I}_{\mathcal{P}}(N_i, N) + \pi_{l+1} \Delta \mathcal{I}_{\mathcal{P}}(Q, N).$$

    Note that if Q is one of the nodes which already propagated their interest to N, the above sum would again sum only l propagated feedback values and $\Delta \mathcal{I}_{\mathcal{P}}(Q, N)$ would be summed with the already accumulated propagated interest value from the node Q. If Q is obtaining the user feedback for the first time, then $\Delta \mathcal{I}_{\mathcal{P}}(Q, N) = \mathcal{I}_{\mathcal{P}}(Q, N)$.

Termination

Regarding termination, for each user feedback, the appropriate portion of the sensed interest is propagated to all the nodes in the graph, but none of the nodes is fired to propagate in the same cycle, as is the case in classical spreading activation. Since we propagate the interests using the precomputed matrix DM, with finite distance between each two pairs of nodes, in each cycle the interest values are updated for all the nodes in the graph (except the node which received the feedback) and loops are avoided. If n is the total number of feedbacks and m is the number of nodes in the graph, then the propagation is executed $n \cdot (m-1)$ times.

Stability

The sensed interest value is calculated according to the Formula 1. Sigmoid function can only have values in the interval $(-1, 1)$ (even though in our evaluation the arguments were only from the interval $[0, 1]$). This means that the value of sensed interest, regardless of the number of feedback values provided by the user can only belong to the interval $(-\frac{\ell(N)}{max}, \frac{\ell(N)}{max})$. Further, the propagated interest value, calculated according to Formula 5, is computed by weighing the sensed interest by a factor smaller then 1, hence again belongs to the same interval. Finally, the constants a, b, c, d, e, f in Algorithm 2 are chosen in such a way to ensure that each weighted sum is again in the given interval.

4.3. Performance

The distance matrix computation procedure is very fast on an average sized ontology. We tested it on the two ontologies used for the evaluation (Section 7) consisting of 26 and 37 classes, yielding runtimes of 2.5 and 4 seconds, respectively. The runtimes for interest propagation were satisfactory. On the ontology with 27 classes, the interest propagation algorithm computed 696 feedback values from 87 users in about 50 seconds, taking 0.07 seconds for every feedback. On the ontology with 37 classes, 2,772 feedback values from 231 users were processed in about 250 seconds, which is 0.09 seconds for each feedback. Note that most of the run time was simply due to the databases queries to retrieve and update the user models. It is possible to improve the performance by optimizing the interaction with the databases or by setting a threshold for the propagation distance, thus propagating the interest only to the nodes that are considered similar enough.

5. Use Case

Let us consider the following scenario. A person is visiting a wine fair in Tuscany and is using the application which permits her to explore the rich realm of Italian wines and also employs our user interests propagation algorithm. The system already knows that she likes Barbera wine, since she commented on it, and this allows the system to give her recommendations based on her initial preference.

In order to illustrate the idea of user interest propagation we would consider the Wine conceptual hierarchy given in Figure 1, which is a part of Drink conceptual hierarchy used in the evaluation process (see Figure 7). In this conceptual hierarchy, the

domain concept *Wine* is a surface node, i.e. its level is equal to zero. We will assume that the level of the deepest node in the hierarchy (calculated starting from the surface node) is $max = 4$. We will assume the following weights for different user actions [13]: commenting $f = 0.5$, bookmarking $f = 0.9$ and tagging $f = 0.7$. Finally, the values of the constants a, b, c, d, e, f are always $\frac{1}{2}$.

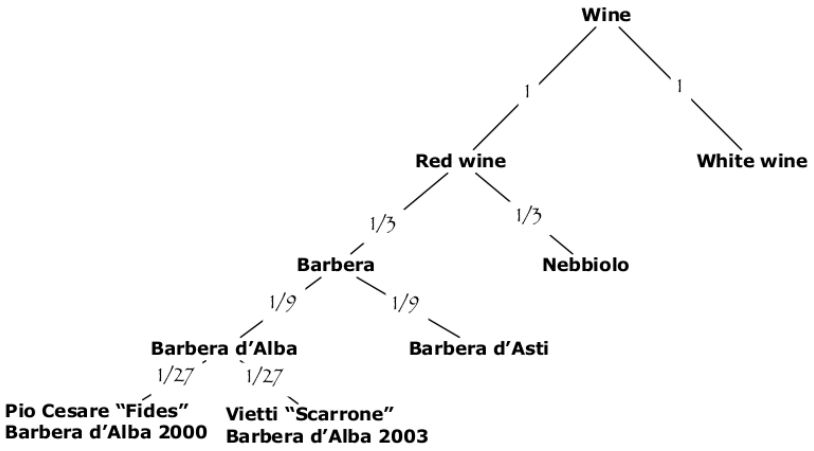

Figure 1: The portion of conceptual hierarchy representing wines

In the example, the user expressed interest in red wine *Barbera*, by commenting on it, which corresponds to the value $f_1 = 0.5$. The level of *Barbera* in the hierarchy is $\ell(Barbera) = 2$. With this information, our system is able to (i) infer the interest value for this class, and (ii) infer the level of interest in the related classes (descendants and ancestors, as well as siblings).

When the node corresponding to *Barbera* receives this first feedback $f_1 = 0.5$, its sensed interest is

$$\mathcal{I}_S(Barbera) = \frac{\ell(Barbera)}{max}\frac{e^{f_1}-1}{e^{f_1}+1} = \frac{1}{2}\frac{e^{0.5}-1}{e^{0.5}+1} = 0.12 \text{ and}$$

$$\mathcal{I}(Barbera) = \frac{1}{2}\mathcal{I}_S(Barbera) = 0.06.$$

This value will be propagated "as is" through the conceptual hierarchy, since there is no previous feedback to be taken into account

$$\Delta\mathcal{I}^1_S(Barbera) = 0.12.$$

When the same node receives the second feedback $f_2 = 0.9$ (the user bookmarks the object), the total sensed interest will become

$$\mathcal{I}_S(Barbera) = \frac{\ell(Barbera)}{max}\frac{e^{f_1+f_2}-1}{e^{f_1+f_2}+1} = 0.3 \text{ and } \mathcal{I}(Barbera) = 0.15$$

The amount of interest that will be propagated to update the system will be

$$\Delta\mathcal{I}^2_S(Barbera) = 0.3 - 0.12 = 0.18.$$

In order to illustrate vertical propagation, we additionally assume that $k = 1$ in equation (3) and $c = 3$ in the calculation of the edge lengths. The choice of c depends on many factors, for example how flat or deep certain sub-ontologies are. Starting from the value $\mathcal{I}_{\mathcal{S}}(Barbera)$ of sensed interest for *Barbera*, we calculated above the variation in the sensed interest $\Delta^i\mathcal{I}_{\mathcal{S}}(Barbera)$, $i \in \{1, 2, 3\}$ for each of the three feedback values provided by the user. These values are propagated further, using the exponentially decreasing edge lengths (see Figure 1). In what follows we use the following abbreviations for better legibility: *Red* for *Red Wine*, *Alba* for *Barbera d'Alba*, *Asti* for *Barbera d'Asti*, *Fides* for *Pio Cesare "Fides" Barbera d'Alba 2000* and *Scarrone* for *Vietti "Scarrone" Barbera d'Alba 2003*.

First of all, we calculate the conceptual distances between the nodes in the part of the hierarchy describing wines.

$$\begin{aligned} dist(Barbera, Wine) &= 1 + \tfrac{1}{3} = \tfrac{4}{3} \\ dist(Barbera, Red) &= \tfrac{1}{3} \\ dist(Barbera, Nebbiolo) &= \tfrac{1}{3} + \tfrac{1}{3} = \tfrac{2}{3} \\ dist(Barbera, Alba) &= dist(Barbera, Asti) = \tfrac{1}{9} \\ dist(Barbera, Fides) &= dist(Barbera, Scarrone) = \tfrac{1}{9} + \tfrac{1}{27} = \tfrac{4}{27}. \end{aligned}$$

In the case of downward propagation, after the first feedback for *Barbera* is obtained and assuming that the descendants and ancestors of *Barbera* did not receive any direct feedback from the user in the past, we have that:

$$\begin{aligned} \mathcal{I}(Alba) &= \tfrac{1}{2}\Delta\mathcal{I}^1_{\mathcal{P}}(Barbera, Alba) = \tfrac{1}{2}e^{-\frac{1}{9}}\Delta\mathcal{I}^1_{\mathcal{S}}(Barbera) = \tfrac{1}{2}0.89 \cdot 0.12 = 0.053 \\ \mathcal{I}(Fides) &= \tfrac{1}{2}\Delta\mathcal{I}^1_{\mathcal{P}}(Barbera, Fides) = \tfrac{1}{2}e^{-\frac{4}{27}}\Delta\mathcal{I}^1_{\mathcal{S}}(Barbera) = \tfrac{1}{2}0.86 \cdot 0.12 = 0.051. \end{aligned}$$

On the other hand, for upward propagation, we obtain:

$$\begin{aligned} \mathcal{I}(Red) &= \tfrac{1}{2}\Delta\mathcal{I}^1_{\mathcal{P}}(Barbera, Red) = \tfrac{1}{2}e^{-\frac{1}{3}}\Delta\mathcal{I}^1_{\mathcal{S}}(Barbera) = \tfrac{1}{2}0.71 \cdot 0.12 = 0.042 \\ \mathcal{I}(Wine) &= \tfrac{1}{2}\Delta\mathcal{I}^1_{\mathcal{P}}(Barbera, Wine) = \tfrac{1}{2}e^{-\frac{4}{3}}\Delta\mathcal{I}^1_{\mathcal{S}}(Barbera) = \tfrac{1}{2}0.26 \cdot 0.12 = 0.015. \end{aligned}$$

Finally, for the horizontal propagation we have:

$$\mathcal{I}(Nebbiolo) = \tfrac{1}{2}\Delta\mathcal{I}^1_{\mathcal{P}}(Barbera, Nebbiolo) = \tfrac{1}{2}e^{-\frac{2}{3}}\Delta\mathcal{I}^1_{\mathcal{S}}(Barbera) = \tfrac{1}{2}0.51 \cdot 0.12 = 0.03.$$

We can see that in the downward propagation the highest value is obtained for the direct descendants, followed by the next descendants in the hierarchy. In the case of upward propagation, the propagated interest for the first antecedent is lower than the value for descendants and rapidly decreasing. This prevents non-proportional propagation upwards, as the specificity of concepts, as well as their importance in the domain, decreases. We can see that the propagated user interest for *Nebbiolo* is smaller with respect to *Red*: this makes sense since *Nebbiolo* is a particular kind of red wine which certain users who like *Barbera* might or might not like. Also, this value is lower than the calculated interest values for specific *Barbera* wines. It simply gives us some estimate of how much the user could like another kind of red wine.

After the second feedback from the user is received for *Barbera*, the appropriate interest values for its descendant and ancestor nodes are the following:

$$\begin{aligned} \mathcal{I}(Alba) &= \tfrac{1}{2}\Delta\mathcal{I}^1_{\mathcal{P}}(Barbera, Alba) + \tfrac{1}{2}\Delta\mathcal{I}^2_{\mathcal{P}}(Barbera, Alba) \\ &= \tfrac{1}{2}e^{-\frac{1}{9}}(\Delta\mathcal{I}^1_{\mathcal{S}}(Barbera) + \Delta\mathcal{I}^2_{\mathcal{S}}(Barbera)) = \tfrac{1}{2}0.89 \cdot (0.12 + 0.18) = 0.133 \end{aligned}$$

$\mathcal{I}(Fides) = 0.129 \qquad \mathcal{I}(Red) = 0.106 \qquad \mathcal{I}(Wine) = 0.039 \qquad \mathcal{I}(Nebbiolo) = 0.076.$

The same procedure would be repeated for the third feedback f_3 for *Barbera*. In addition, in case some of these nodes receives a direct feedback from the user, this value would be added to the total value of user interest for that item.

Of course, by using different values for k and c we can model different propagation behaviors. We can see that, as the user simply expresses his interest for *Barbera*, the system starts to infer his interest in all the related classes. After a little feedback the application will be able to suggest to the user other objects he probably likes. For example, it will know that he prefers red wine to white wine.

6. Support for dynamic ontology maintenance

In many real systems the domain ontology is not immutable, actually there might exist many reasons to modify it once it is already in use. For example, after some time certain classes may not categorize the domain concepts precisely enough, thus requiring modifications, or a certain category may need to be abandoned or added. This is the issue of *ontology maintenance* [9, 8], i.e. the ability to automatically evaluate and, in case it is needed, add new classes in the taxonomical structure in order to improve the categorization. Our approach supports the ontology maintenance, not in the sense of correcting the taxonomy by adding or removing classes, but allowing to update the values after the modification happens. Thus, after changing the conceptual hierarchy, our approach supports the update of the interest values for all the domain objects as if the system had always used the modified conceptual hierarchy.

When a leaf node N needs to be removed from the conceptual hierarchy, it should be possible to nullify the effect that the leaf had on the system. From formula 3 we can obtain the exact amount of interest propagated from the node N to every other node in the conceptual hierarchy. Hence, we can delete the effect of the past propagations from the node N by propagating the value $-\mathcal{I}_{\mathcal{P}}(N, M)$ to each node M.

When a leaf node N needs to be added to the conceptual hierarchy, it is possible to propagate the interest from all the nodes in the conceptual hierarchy, as if that node had existed since the beginning. To achieve this, every node M should propagate to the new node N the amount of interest $\mathcal{I}_{\mathcal{P}}(M, N)$ using again formula 3. We can limit this propagation to the nodes whose distance from N is smaller than $maxP$, in case we want to avoid propagation to the nodes which only have a surface node as a common ancestor with N (see end of Section 3).

If we need to remove or add a node which is not a leaf node, we must take into consideration the fact that the distances between other nodes will change. With respect to the portion of the ontology representing wines, given in Figure 1, the removal of a non-leaf node is illustrated in Figure 2, whereas the addition of a non-leaf node is illustrated in Figure 3. In the first case, the node *Barbera* is removed, changing all the distances apart from $dist(Wine, White)$, $dist(Wine, Red)$ and $dist(Red, Nebbiolo)$. In the second case, the node *Barbera Piemonte* is added, changing again all the distances apart from the above mentioned ones.

In case of the removal of the node N, the procedure is the following:

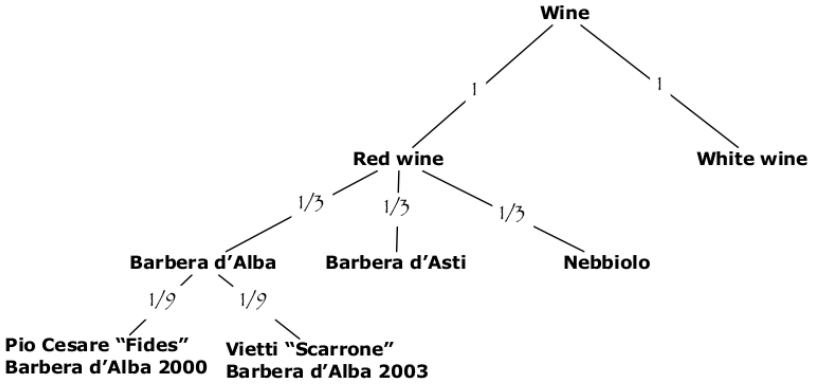

Figure 2: The portion of domain ontology representing wines with *Barbera* node removed

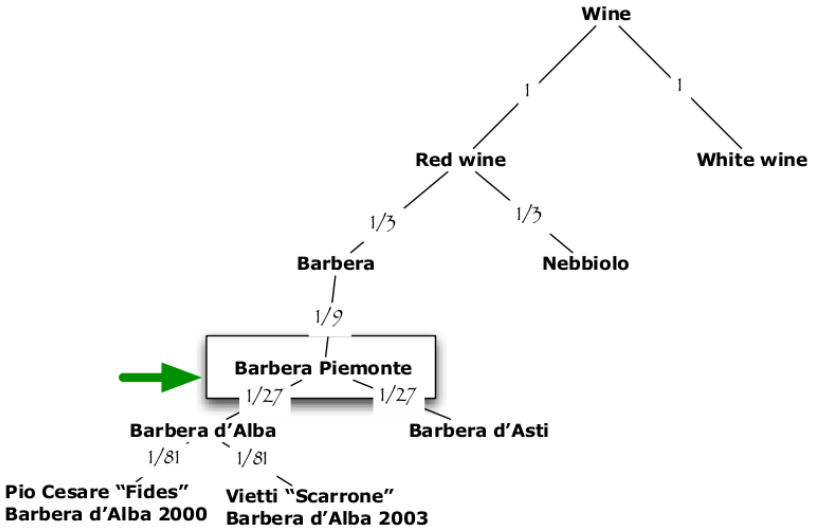

Figure 3: The portion of domain ontology representing wines with *Barbera Piemonte* node added

- Cancel the effect of the previous propagation of interests from the node N by propagating $-\mathcal{I}_{\mathcal{P}}(N, M)$ to each node M in the conceptual hierarchy.
- Modify the conceptual hierarchy by deleting the node N.
- Re-calculate the distances in the modified conceptual hierarchy.
- Re-propagate the interest $\mathcal{I}_{\mathcal{P}}(M, Q)$ from each node M to each node Q in the modified conceptual hierarchy, for which the distances have changed.

In case of the addition of the node N, the procedure is the following:

- Modify the conceptual hierarchy by adding the node N.
- Re-calculate the distances in the modified conceptual hierarchy.
- Re-propagate the interest $\mathcal{I}_{\mathcal{P}}(M, Q)$ from each node M to each node Q in the modified conceptual hierarchy, for which the distances have changed.

In the previous version of our approach [14] it was not possible to support the change of the ontology and one of the reasons for improving our propagation mechanism was to overcome this limitation.

7. Evaluation

In order to evaluate our approach, we used an adaptive application *WantEat*, developed as a part of the PIEMONTE Project [17]. The application is placed in the domain of gastronomy, where intelligent objects such as products (cold cuts, wines, cheeses), as well as shops, restaurants, market stalls etc. provide information and recommendations to users. To represent the domain, we used a domain ontology where additional sub-ontologies describing particular areas of the domain are imported (products, recipes, actors, places). The domain objects are modeled as instances of the ontology classes.

The most important personalization aspect of the application is the personalized ranking of the the objects returned by a search or displayed during navigation where it is of utter importance to mirror the users preferences stored in the user model.

We tested the algorithm by running two evaluations on two different conceptual hierarchies: the portion of domain ontology representing cold cuts (Figure 4) and the one regarding drinks (Figure 7), since they constitute a well-known domain also for people not expert in food domain, and they are fairly well balanced and easy for users to provide feedback on. In both experiments, we empirically set $c = 3$ when calculating the edge lengths (see Section 3.1).

7.1. Hypothesis

Our assumption was that our algorithm can be used to predict users interests in certain objects of a given domain. In particular, we wanted to measure the accuracy of our prediction, showing that the lists of domain items generated by our algorithm provide users with recommendations in good correlation with the lists of liked items provided by the users themselves.

Furthermore, we wanted to prove that taking into consideration both the horizontal *and* vertical propagation provides better results than using only vertical propagation, as it was the case in our previous work [14] and other similar state of the art approaches [30, 39] that will be discussed in Section 8. In this evaluation we use the term *Multi-Directional Approach* (MDA) for the new approach, which takes into account

both horizontal and vertical propagation, as opposed to the baseline *Vertical Approach* (VA) used previously. Moreover, in the last test, we also compared the new *Anisotropic Propagation* (see Section 3.1) based on conceptual distance with the old method which uses uniform distances when calculating the distance between two nodes, in order to show that the introduction of conceptual distance improves the propagation results.

7.2. Measures

The correlation between the list generated by the tested algorithm and the original user's list was estimated using Spearman's rank correlation coefficient ρ, which provides a non-parametric measure of statistical dependence between two ordinal variables and gives an estimate of the relationship between two variables using a monotonic function.

In the absence of repeated data values, a perfect Spearman correlation of $\rho = +1$ or $\rho = -1$ is obtained when each of the variables is a prefect monotone function of the other. Ties values are assigned a rank equal to the average of their position in their descending order of the values. Usually $\rho \geq 0.5$ points to a "fair direct correlation", $\rho \geq 0.7$ to a "good direct correlation" and $\rho \geq 0.9$ to a "very good direct correlation".

The performance of the different techniques were also compared with the chi-square test, to check whether the differences were statistically significative.

7.3. Cold Cuts test

In order to collect users' preferences regarding some domain objects, we used a two-part questionnaire. In the first part, the users were asked to rank their preferences for 8 non-leaf classes of our hierarchy (e.g. *Raw_Whole_ColdCuts*) on a scale from 1 to 10. In the second part, they were asked to indicate 4 leaf classes (e.g. *Raw_Ham*, *Mortadella* etc.) they "liked very much" and 4 leaf classes they "liked enough" among the total of 15 leaf classes. All the users were instructed to connect to the web site with written instructions and compile the anonymous two-part questionnaire. Users' ratings were registered in a database. This permitted us to collect users' preferences divided in two parts, in order to test upward, downward and horizontal propagation.

7.3.1. Subjects

The initial sample included 100 subjects, 19-45 years old, recruited according to an availability sampling strategy[7]. We restricted the initial sample to 87 subjects by eliminating all the subjects who assigned the same vote to 5 out of 7 non-leaf classes, since this was a strong indicator of random preferences assignment by these users.

7.3.2. Test description

We distinguished two phases in the evaluation process:

[7] Even though the best way for having a representative sample is random sampling, these strategies are time consuming and not always financially justifiable. Therefore, much research in social science is based on samples obtained through non-random selection, such as the availability sampling, i.e. a sampling of convenience, based on subjects available to the researcher, often used when the population source is not completely defined.

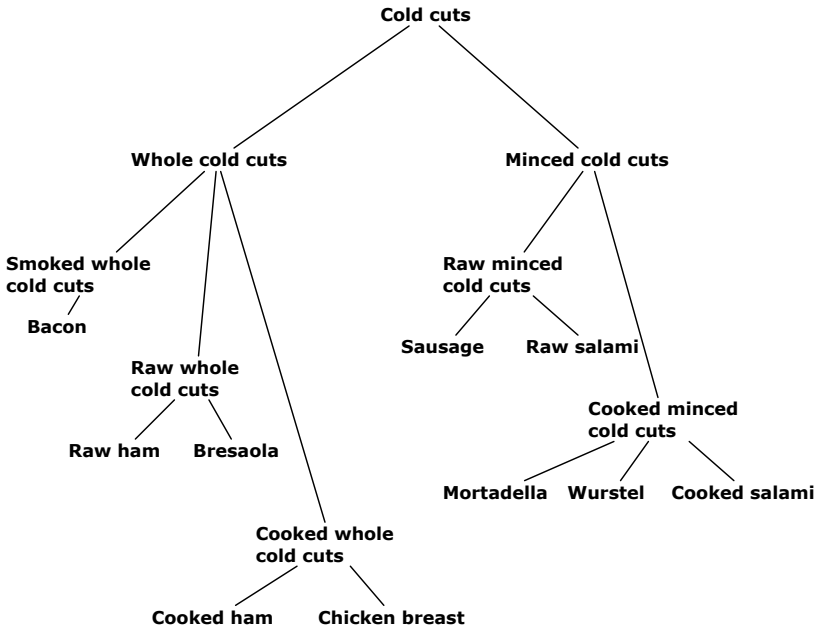

Figure 4: The portion of the conceptual hierarchy representing cold cuts

- *Upward propagation* evaluation, using both the MDA and the VA. The collected 8 user feedback values for the leaf classes for each user were used to generate a list of direct parent classes that was then compared with the original list provided by the user.
- *Downward propagation* evaluation, using again both MDA and VA. The collected 7 user feedback values for the non-leaf classes for each users were used to generate a list of leaf classes which were then compared with the original list provided by the user.

7.3.3. Results

We collected a total of 1392 ratings and calculated Spearman's rank correlation coefficient ρ for 87 pairs of lists. The graphs shown in Figures 5 - 6 refer to the distribution of cases for various values of ρ. For example, the percentage of cases corresponding to 0.8 on the ρ axis is the fraction of cases which obtained $0.75 < \rho \leq 0.85$.

a) *Upward propagation.*

Figure 5 shows that the MDA outperforms the VA yielding 92% of the cases with a positive association as opposed to 62% in the VA. Moreover, it leads to 32% of cases with a "good correlation" ($\rho \geq 0.7$) as opposed to 12% in the VA. The number of cases with $\rho \leq -0.5$ is as low as 2% for the MDA, whereas it is 13% for the VA. The difference between performances were statistically significant according to the chi-square test, with a null hypothesis $p = 6 \cdot 10^{-12}$.

b) *Downward propagation.*

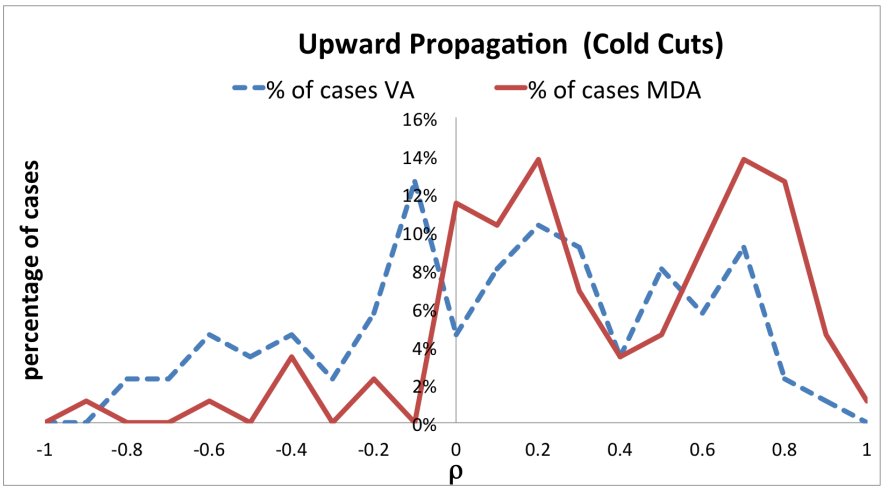

Figure 5: Cold Cuts: distribution of cases for various values of ρ for the upward propagation. A comparison between the MDA and VA approaches.

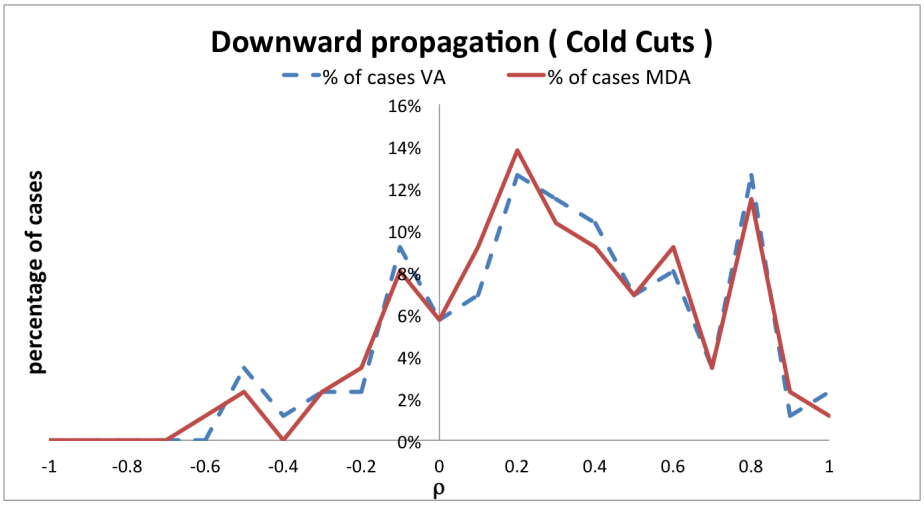

Figure 6: Cold Cuts: distribution of cases for various values of ρ for the downward propagation.

In this case the results of MDA and VA do not differ much. We were able to generate a list with a positive association in 82% of the cases with MDA and in 81% of the cases with the VA. In particular, in 34% of the cases with both approaches we were able to generate the lists with a fair association ($\rho \geq 0.5$) and in 18% for MDA (19% VA) of the cases a list with a good association ($\rho \geq 0.7$). We had 3% of cases of misleading lists with a moderate inverse association ($\rho \leq -0.5$) for both approaches. This is the only test in which the two approaches do not yield statistically significant differences in performance.

7.4. Drinks test

As before the user preferences were split into two sets to serve as initial feedback and ground truth. We computed on this basis the propagation and ρ for the ordered lists of items. In this case we had many low-level leaves, and thus we tested also the leaf to leaf propagation and the mixed propagation in which we propagated from leaf and from non-leaf classes at the same time. Since this test is more general and complex, we decided to use it also for evaluating the benefits of using an anisotropic technique based on the conceptual distance.

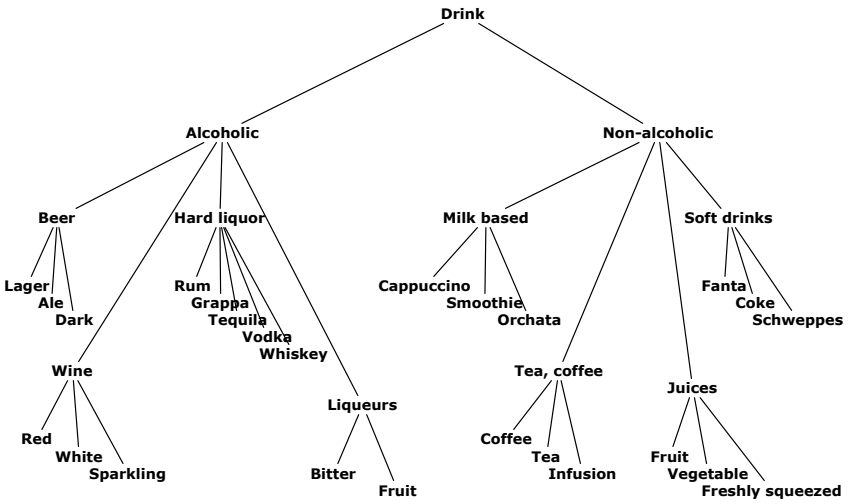

Figure 7: The portion of the conceptual hierarchy representing drinks

7.4.1. Subjects

The initial sample included 240 subjects, 19-38 years old, recruited as in the Cold Cuts test. As before we restricted the initial sample to 231 subjects by eliminating all the subjects who assigned the same vote to 25 out of 35 classes, since this was a strong indicator of random preferences assignment by these users.

7.4.2. Test description

We distinguished four phases in the evaluation process:

- *Upward propagation*, where using the same procedure as before we compared the users' real feedback for leaf classes to the computed list from non-leaf classes (24 feedback values for each user).
- *Downward propagation* where, as before, we compared the users' real feedback for non-leaf classes to the list computed from leaf classes (10 feedback values for each user).
- *Leaf to leaf propagation*, where we split randomly the user feedback values on the leaves into two sets of 12 classes and computed the Spearman coefficient between the two.
- *Mixed propagation*, where we compared the users' real feedback for both non-leaf and leaf classes to the list computed from both non-leaf and leaf classes. We split the user feedback values in two sets of 12 leaf and 5 non-leaf classes each and computed the Spearman coefficient between the two.

7.4.3. Results

For this test we collected a total of 7854 ratings and calculated Spearman's rank correlation coefficient ρ for 231 pairs of lists.

a) *Upward propagation*

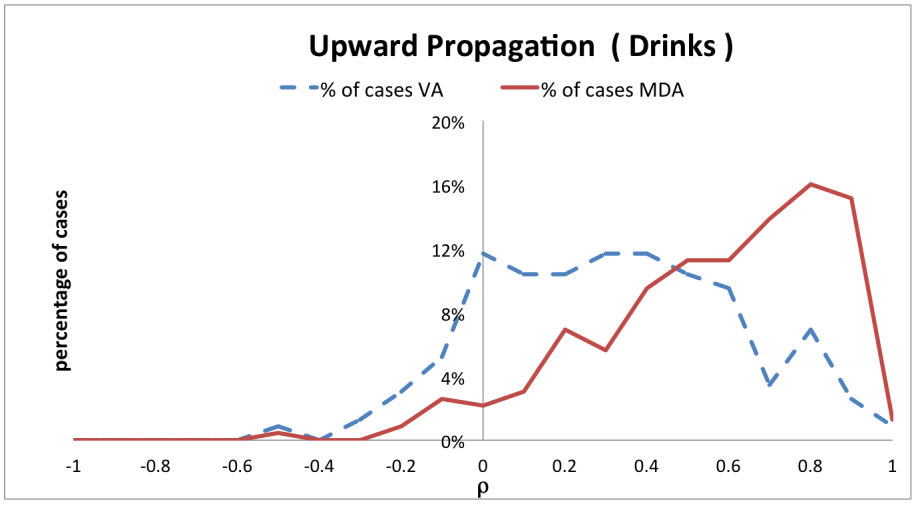

Figure 8: Drinks: distribution of cases for various values of ρ for the upward propagation. A comparison between the MDA and the VA

We can see in Figure 8 that the shapes of the curves obtained for upward propagation are similar to the ones obtained for the Cold Cuts test, but the results are much better due to the higher number of feedback values (24 instead of 8).

We were able to obtain a list with a positive correlation in 96% of the cases for MDA (90% VA). Moreover, we obtained a list with $\rho \geq 0.7$ in 46% of the cases for MDA (14% VA). In almost one fifth of the cases we suggested a list with a strong

association ($\rho \geq 0.9$). It appears that, as the amount of feedback grows, the technique becomes more reliable and in fact only in one case out of 231 we got a misleading list ($\rho \leq -0.5$). The performance improvement caused by the MDA is statistically significative ($p = 10^{-200}$) .

b) *Downward propagation*

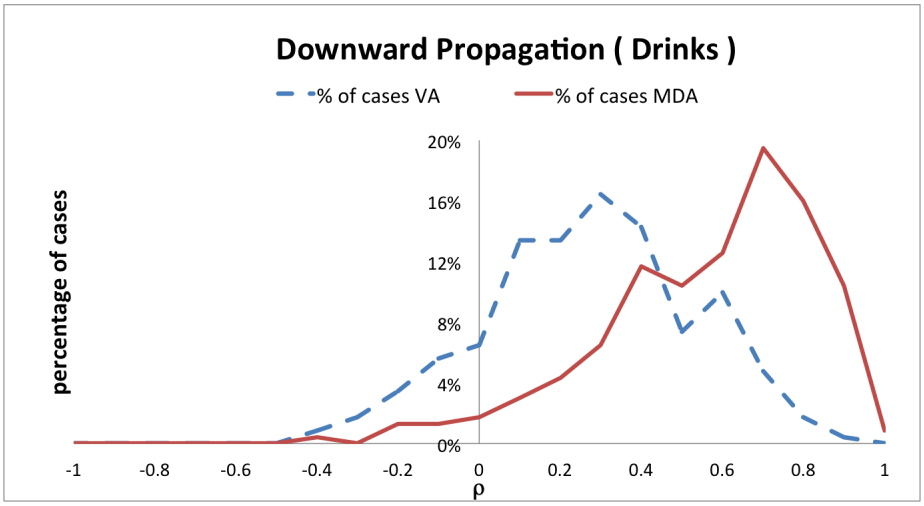

Figure 9: Drinks: distribution of cases for various values of ρ for the downward propagation. A comparison between the MDA and the VA.

As shown in Figure 9, in this case the MDA seems to work better for the downward propagation. In 97% of the cases for MDA (87% VA) the lists had a positive association. As in the upward propagation we obtained a good association in 47% of the cases for MDA (7% VA). No misleading lists were obtained. The difference between the two approaches is statistically significative with $p = 5 \cdot 10^{-54}$.

c) *Leaf to leaf propagation*

As shown in Figure 10 the leaf to leaf propagation gave results similar to those obtained by using the upward and downward propagation. We recorded 92% of the cases with a positive association and 36% with a good association ($\rho \geq 0.7$). In 9% of the cases we were able to generate a list with $\rho \geq 0.9$ whereas less then 1% of the cases fell under $\rho \leq -0.5$.

d) *Mixed propagation*

In this part of the test, we wanted to compare the anisotropic MDA, the non-anisotropic MDA and the simplest case, the pure VA. As shown in Figure 11, the MDA performs always much better than the VA. This is an expected outcome.

Moreover, the anisotropic MDA, used in the previous tests, works better than the non-anisotropic version of the MDA, yielding both a higher percentage of cases with a positive association and a lower percentage of errors. The former obtained a fair association in 61% of the cases, whereas the non-anisotropic one in the 51%. It also yields a strong association ($\rho \geq 0.7$) in 34% of the cases as opposed to 28% in the non-anisotropic case. The differences between the VA and the not-anisotropic MDA

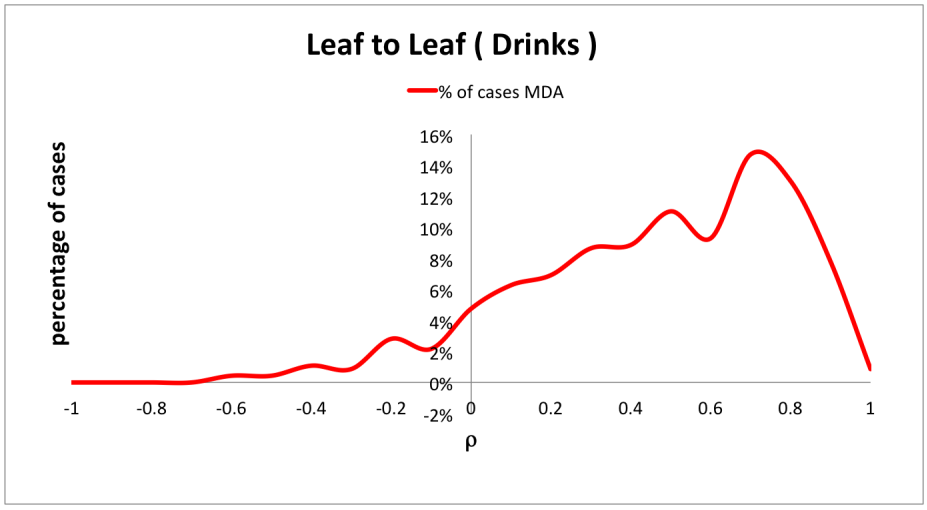

Figure 10: Drinks: distribution of cases for various values of ρ for the leaf to leaf propagation.

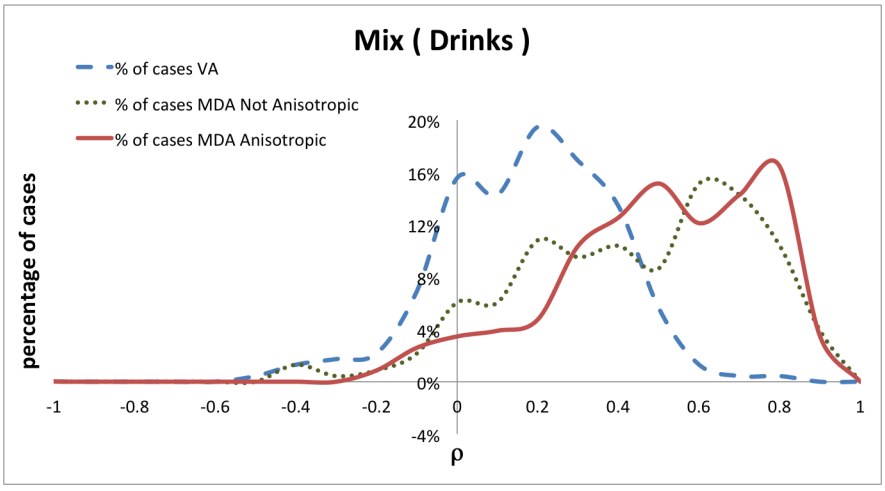

Figure 11: Drinks: distribution of cases for various values of ρ for the mixed propagation. A comparison between the anisotropic MDA, the non-anisotropic MDA and the pure VA.

are significative ($p < 10^{-200}$) , and so are the differences between the not-anisotropic MDA and the anisotropic MDA ($p = 2 \cdot 10^{-4}$). This confirms that, while a MDA works better than the classical VA, the anisotropic version of the MDA is even more effective.

7.5. Discussion

The tests and their statistical significance (via chi-square test) are summarized in Table 1. All differences between the two approaches were found statistically significant, with exception of the downward propagation for cold cuts.

	Cold Cuts		Drinks		
	Upward	Downward	Upward	Downward	Mix
	r < 0				
VA	37.9% (28.2-48.5%)	18.4% (11.3-27.6%)	10.4% (6.9-14.8%)	11.7% (8.0-16.3%)	12.6% (8.7-17.3%)
MDA	**8.0%** (3.6-15.3%)	**17.2%** (10.4-26.3%)	**3.9%** (1.9-7.0%)	**3.0%** (1.3-5.9%)	**3.5%** (1.6-6.5%)
	r ≥ 0.7				
VA	12.6% (6.8-20.9%)	**19.5%** (12.2-28.9%)	13.9% (9.8-18.8%)	6.9% (4.1-10.8%)	0.9% (0.2-2.8%)
MDA	**32.2%** (23.0-42.5%)	18.4% (11.3-27.6%)	**46.3%** (40.0-52.8%)	**46.8%** (40.4-53.2)	**34.2%** (28.3-40.5)
	Chi-square test				
VA vs MDA	$p=6 \bullet 10^{-12}$	p=0.99	$p=5 \bullet 10^{-54}$	$p<10^{-200}$	$p<10^{-200}$

Table 1: VA vs MDA: summary of the main results of the evaluation (in parenthesis the 95% confidence interval). In bold the best result of each test (less is better for $\rho < 0$, more is better for $\rho \geq 0.7$).

The results show that both downward and upward propagation work well with two different conceptual hierarchies, allowing to generate lists with a high degree of association with the user preferences. The MDA works better than the VA in all the tests except for the downward propagation for cold cuts, where they fared in a similar way. The Drinks test which used a total of 34 feedback values instead of 15 feedback values for each user led to much better results, especially for the MDA. In particular, the good correlation cases ($\rho \geq 0.7$) increased from 14% to 46% for the upward propagation and from 7% to 47% for the downward propagation. Both the leaf to leaf and the mixed propagation worked satisfactorily, yielding a list with a good association in more then one third of the cases. Moreover the anisotropic MDA shows a significant improvement with respect to the non-anisotropic MDA.

The misleading cases in which the suggested lists had a moderate inverse association with the real preferences ($\rho \leq -0.5$) were very low (3% - 5%) in the first test and less then 1% in the second. In all cases the MDA obtained a lower percentage of misleading cases. Thus this approach appears to yield a lower risk of misinterpreting the user preferences.

8. Comparative analysis of related work

In this section we provide the comparative analysis of our approach with similar approaches in the literature. Two main features of our approach are:

- usage of ontological-user model;
- exploitation of spreading activation technique in order to propagate user interests in the conceptual hierarchy.

Hence, we look into the related work that deals with these two features. In Section 8.1 works that employ both features are analyzed. Section 8.2 deals with works that simply use ontologies to represent user models but do not propagate interest values on the corresponding ontologies. Finally, Section 8.3 is reserved for works that use spreading activation but out of the context of user modeling.

8.1. Related work exploiting ontology-based user model and ontology propagation

The most similar related works are Sieg et at. [39, 40] and Middleton et al. [30]. All use the ontology based user model and propagate user interests in the ontology.

In Sieg et al. [39], ontological user profiles are used to improve personalized Web search. The user profiles are instances of a reference domain ontology, where user profiles are incrementally updated based on the user interaction with the system. The weights for the relations between each concept and all of its subconcepts are computed using a measure of containment, whereas in our case they are calculated using the conceptual distance [16]. The concept hierarchy is treated as a semantic network and interest values are updated by spreading activation. The amount of activation that is propagated to each neighbor is proportional to the weight of the relation. This results in propagation of the values only in one direction, whereas we perform a multi-directional propagation of the values. In addition, they neither take into consideration the amount and type of user feedback received for each node, nor the level of the node in the conceptual hierarchy.

In Sieg et al. [40], the authors exploit the same propagation approach in a different context, namely collaborative-filtering recommendation. They consider the relationships among concepts in the ontology as well as the collaborative evidence derived from the ontological profiles of similar users. User similarities are computed based on their interest scores for ontology concepts, instead of their ratings of individual items.

In Middleton et al. [30], the authors try to increase the accuracy of the user profiles and hence usefulness of the recommendations in hybrid recommender systems. The user feedback is collected and mapped onto the domain ontology. The IS-A hierarchy is exploited to infer additional topics of interest (superclass topics starting from specific topics browsed by a user), thus producing more complete profiles. The propagation of interest values in the taxonomy is based on IS-A similarity. As opposed to us, their propagation is static and one-directional (bottom-up, from specific to generic concepts): given the interest value for a specific class, 50% of this value is spread to its super class. Another difference is provided by their usage of a time decay function which gives more weight to the recently seen concepts than to the older ones, while in our approach past interaction is more important.

Table 2 summarizes distinct and common features of these approaches, emphasizing the most relevant features: User model definition, Propagation modality, Propagation direction, Concept similarity measure, Propagation target, Time, User feedback, Ontology level and Application domain.

Features	OBUM	Sieg et al. [40], [39]	Middleton et al. [30]
User model	taxonomy-based	taxonomy-based	taxonomy-based
Propagation modality	anisotropic spreading activation based on conceptual distance	uniform spreading activation	anisotropic IS-A-based propagation based on cluster, specificity
Propagation direction	multi-directional (siblings, ancestors, descendants)	one-directional (top-down)	one-directional (bottom-up)
Concept similarity	conceptual distance	measure of containment	IS-A-similarity
Propagation target	classes and instances	classes (starting from instances)	classes (starting from instances)
Time	past more important	no time importance	present more important
User feedback	yes (different weights)	yes (all equals)	yes (all equals)
Ontology level	yes (different weights)	no (all equals)	no (all equals)
Application domain	adaptive systems	recommendation, Web search	recommendation

Table 2: Comparison with the most similar related work

8.2. Works exploiting ontology-based user model

For many years, ontology-based user models have been recognized as useful for adaptive systems, as discussed in Section 2. As examples we can cite Kalfoglou et al. [24], Gauch et al. [19] and Hatala and Wakkary [21]. These works do not perform spreading activation over the ontology, they rather make other forms of inferences in order to infer missing values in the user model.

In MyPlanet [24], a recommender system for scientific information, user preferences are mapped to a set of seven ontologies representing research areas, research themes, organizations, projects, etc. There is no weight associated with a preference, rather it is a binary value. MyPlanet uses ontological inference to propagate user preferences throughout an ontology and across different ontologies: starting from a user query about a concept, it presents also related concepts.

Gauch et al. [19] explore techniques for implicit building of ontology-based user profiles, starting from visited Web pages. They calculate the similarity between the Web pages visited by a user and the concepts in a domain ontology. After annotating each concept with a weight based on an accumulated similarity score, a user profile is created consisting of all concepts with non-zero weights. They demonstrated an improvement in the user model accuracy when it is supported by the use of a domain ontology.

In ec(h)o, a museum guide presented by Hatala and Wakkary [21], user interests are represented as a set of annotated concepts from the ontology of natural history. The strength of user interest is represented by a weight calculated based on the user activities in the physical museum place. In this model, the recent activities weigh more

than past ones. When the interest value drops below a set threshold during the update process the interest is removed from the model altogether.

8.3. Works using spreading activation

Spreading activation is not a new concept in semantic networks related research. There is a number of applications of spreading activation: Crestani [18] uses spreading activation on automatically constructed hypertext networks in order to support web browsing; Xue et al. [44] propose a mining algorithm to improve web search performing spreading activation to re-rank the search results; Liu et al. [27] use spreading activation on a semantic network of automatically extracted concepts in order to identify suitable candidates for expanding a specific domain ontology; Katifori et al. [25] use spreading activation through ontologies to support personal information management.

In Rocha et al. [35] spreading activation techniques are used to find related concepts in the ontology given an initial set of concepts and corresponding initial activation values. They assign both a semantic label and a numerical weight based on certain properties of the ontology to each relation instance. This technique makes it possible to find concepts not containing any of the words specified in the initial search. Similarly to us, depending on the context, some relation types can be more important than others. Differently from us, they exploit a full ontology with IS-A relations and properties and then make the propagation considering a property-based similarity among concepts.

Another interesting work is ONTOCOPI (Ontology-Based Community of Practice Identifier) [3], a relationship analysis tool, used to identify communities of practice by applying a set of ontology-based network analysis techniques. It examines the connections of the instances with respect to the type, density, and weight. The set of close instances are identified and ranked according to the weight of their relationships. A breadth-first spreading activation search is applied to the ontology, traversing the semantic relationships between instances until a defined threshold is reached.

9. Conclusions and future work

In this work we presented an effective approach to propagation of user interests in an ontology-based user model. Starting from the feedback obtained for a certain node in conceptual hierarchy, it is possible to propagate user interests:

- vertically, the propagation being anisotropic with respect to downward and upward propagation using exponentially decreasing edge lengths from [16];
- horizontally to sibling nodes.

The conceptual hierarchy of the domain plays an important role in the propagation, since propagated interest values depend on the level of the node receiving the feedback. As a consequence, the nodes lower down in the hierarchy can sense and propagate more user feedback. Moreover, in this work we introduced the context-dependent propagation, which allows to take the application context into account in the propagation algorithm (i.e. changing the surface nodes based on their relevance for the application).

This kind of ontological user modeling and maintenance of user interest values finds application in many different contexts, among which to improve Web search results or recommendation accuracy.

Our evaluation results showed that our *multi-directional* approach outperforms the *vertical* one yielding a 12% average improvement for the cases with a positive association and 24% average improvement for the cases with a "good correlation" ($\rho \geq 0.7$). A higher number of feedback values emphasizes this difference: in fact we obtained a 35% average improvement for the cases with a "good correlation" when taking in consideration only the second test.

These positive test results are even more promising in the light of the fact that our method needs only the feedback from a single user for calculating the interest values, whereas other methods need a lot of feedback from many different users to work properly.

On the other hand, our approach still has some open issues which we intend to address in the future.

Very often it can be difficult to establish the correspondence among the users' way of thinking and their level of expertise and the experts' views on the ontology (the ones that actually designed and built the ontology). These possible mismatches can affect the expression of real user interest, i.e. the user feedback does not represent real user interest. As the first future step we would like be able to provide personalised view of the ontology based on the user's level of expertise and the context in which the application is used. For example, offering a number of expertise levels to chose from, would make the ontological concepts closer to the users and make them more comfortable using the application.

Also, since all the siblings have the same distances from their parent, equal amount of user interest is propagated to each of them. In this line, we would design an approach which distinguishes various siblings based on some further differentiating properties.

In addition to propagation of interests based on distances in the ontology graph, we considered the properties of various classes in order to better differentiate the siblings and infer the similarities between them [15]. We are planning to put together the two different approaches, but the integration is not trivial and it needs further investigation.

Taking properties into account would extend our approach to ontologies with implicit classes, i.e. classes not explicitly defined which can be inferred from the structure of of the ontology by some inferential engine.

Moreover, we want to add a mechanism for automatic ontology maintenance, i.e. for refining or evolving the ontology by making use of top-down or bottom-up approaches to concept discovery, as in [9].

Another possible future direction is to exploit ontological user profile and our propagation of interests to compute users' similarity, as in [40].

In addition, we would like to explore adding constraints to our propagation algorithm along the lines of [35], such as concept type constraints (no propagation to certain kinds of nodes).

References

[1] Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17 (6), 734–749.

[2] Akiba, T., Tanaka, H., 1994. A bayesian approach for user modeling in dialogue systems. In: 15th Conference on Computational linguistics - Volume 2. COLING '94. Association for Computational Linguistics, pp. 1212–1218.

[3] Alani, H., Dasmahapatra, S., O'Hara, K., Shadbolt, N., 2003. Identifying communities of practice through ontology network analysis. IEEE Intelligent Systems 18 (2), 18–25.

[4] Allemang, D., Hendler, J., 2008. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann Publishers.

[5] Antoniou, G., Baldoni, M., Baroglio, C., Patti, V., Baumgartner, R., Eiter, T., Herzog, M., Schindlauer, R., Tompits, H., Wien, T. U., Bry, F., Henze, N., 2004. Reasoning methods for personalization on the semantic web. Annals of Mathematics, Computing and Teleinformatics 2, 2004.

[6] Antoniou, G., van Harmelen, F., 2008. A Semantic Web Primer, second edition. The MIT Press.

[7] Berkovsky, S., Kuflik, T., Ricci, F., 2009. Cross-representation mediation of user models. User Modeling and User-Adapted Interaction 19 (1-2), 35–63.

[8] Beydoun, G., 2009. Formal concept analysis for an e-learning semantic web. Expert System Application 36 (8), 10952–10961.

[9] Beydoun, G., Lopez-Lorca, A. A., García-Sánchez, F., Martínez-Béjar, R., 2011. How do we measure and improve the quality of a hierarchical ontology? Journal of System Software 84 (12), 2363–2373.

[10] Brusilovsky, P., 2007. Adaptive navigation support. In: The Adaptive Web, Methods and Strategies of Web Personalization. Vol. 4321 of LNCS. Springer, pp. 263–290.

[11] Brusilovsky, P., Millán, E., 2007. User models for adaptive hypermedia and adaptive educational systems. In: The Adaptive Web, Methods and Strategies of Web Personalization. Vol. 4321 of LNCS. Springer, pp. 3–53.

[12] Bunt, A., Carenini, G., Conati, C., 2007. Adaptive content presentation for the web. In: The Adaptive Web, Methods and Strategies of Web Personalization. Vol. 4321 of LNCS. Springer, pp. 409–432.

[13] Carmagnola, F., Cena, F., Console, L., Cortassa, O., Gena, C., Goy, A., Torre, I., Toso, A., Vernero, F., 2008. Tag-based user modeling for social multi-device adaptive guides. User Modeling and User-Adapted Interaction 18 (5), 497–538.

[14] Cena, F., Likavec, S., Osborne, F., 2011. Propagating user interests in ontology-based user model. In: Advances in Artificial Intelligence, AI*IA '11. Vol. 6934 of LNCS. Springer, pp. 299–311.

[15] Cena, F., Likavec, S., Osborne, F., 2012. Property-based interest propagation in ontology-based user model. In: 20th Conference on User Modeling, Adaptation, and Personalization, UMAP 2012. Vol. 7379 of LNCS. Springer, pp. 38–50.

[16] Chiabrando, E., Likavec, S., Lombardi, I., Picardi, C., Theseider Dupré, D., 2011. Semantic similarity in heterogeneous ontologies. In: 22nd ACM Conference on Hypertext and Hypermedia, Hypertext '11. ACM, pp. 153–160.

[17] Console, L., Lombardi, I., Picardi, C., Simeoni, R., 2011. Toward a social web of intelligent things. AI Communications 24 (3), 265–279.

[18] Crestani, F., 1997. Application of spreading activation techniques in information retrieval. Artificial Intelligence Review 11 (6), 453–482.

[19] Gauch, S., Chaffee, J., Pretschner, A., 2003. Ontology-based personalized search and browsing. Web Intelligence and Agent Systems 1, 1–3.

[20] Gruber, T. R., Jun. 1993. A translation approach to portable ontology specifications. Knowledge Acquisition Journal 5 (2), 199–220.

[21] Hatala, M., Wakkary, R., 2005. Ontology-based user modeling in an augmented audio reality system for museums. User Modeling and User-Adapted Interaction 15 (3-4), 339–380.

[22] Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M., 2005. GUMO - the General User Model Ontology. In: 10th International Conference on User Modeling, UM '05. Vol. 3538 of LNCS. Springer, pp. 428–432.

[23] Ingerman, P. Z., 1962. Algorithm 141: Path matrix. Communications of the ACM 5 (11), 556.

[24] Kalfoglou, Y., Domingue, J., Motta, E., Vargas-Vera, M., Shum, S. B., 2001. MyPlanet: an ontology-driven web-based personalised news service. In: Workshop on Ontologies and Information Sharing at IJCAI '01.

[25] Katifori, A., Vassilakis, C., Dix, A. J., 2010. Ontologies and the brain: Using spreading activation through ontologies to support personal interaction. Cognitive Systems Research 11 (1), 25–41.

[26] Kobsa, A., Koenemann, J., Pohl, W., 2001. Personalized hypermedia presentation techniques for improving online customer relationship. The Knowledge Engineering Review 16(2), 111–155.

[27] Liu, W., Weichselbraun, A., Scharl, A., Chang, E., 2005. Semi-automatic ontology extension using spreading activation. Journal of Universal Knowledge Management 0 (1), 50–58.

[28] Mehta, B., Niederée, C., Stewart, A., Degemmis, M., Lops, P., Semeraro, G., 2005. Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In: 10th International Conference on User Modeling, UM '05. Vol. 3538 of LNCS. Springer, pp. 119–123.

[29] Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S., 2007. Personalized search on the world wide webt. In: The Adaptive Web, Methods and Strategies of Web Personalization. Vol. 4321 of LNCS. Springer, pp. 195–230.

[30] Middleton, S. E., Shadbolt, N. R., De Roure, D. C., 2004. Ontological user profiling in recommender systems. ACM Transactions on Information Systems 22, 54–88.

[31] Mizoguchi, R., Bourdeau, J., 2000. Using ontological engineering to overcome common AI-ED problems. International Journal in Education 11 (2).

[32] Rada, R., Mili, H., Bicknell, E., Blettner, M., 1989. Development and application of a metric on semantic nets. IEEE Trans. on Systems Management and Cybernetics 19 (1), 17–30.

[33] Razmerita, L., Angehrn, A., Maedche, A., 2003. Ontology-based user modeling for knowledge management systems. In: 9th International Conference on User modeling, UM '03. Vol. 2702 of LNCS. Springer, pp. 213–217.

[34] Resnik, P., 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11, 95–130.

[35] Rocha, C., Schwabe, D., Aragao, M. P., 2004. A hybrid approach for searching in the semantic web. In: 13th International Conference on World Wide Web. WWW '04. ACM, pp. 374–383.

[36] Salton, G., Buckley, C., 1988. On the use of spreading activation methods in automatic information. In: 11th Annual International ACM SIGIR Conference on Research and development in information retrieval. SIGIR '88. ACM, pp. 147–160.

[37] Salton, G., McGill, M., 1984. Introduction to Modern Information Retrieval. McGraw-Hill Book Company.

[38] Shapira, B., Rokach, L., Freilikhman, S., 2013. Facebook single and cross domain data for recommendation systems. User Modeling and User-Adapted Interaction 23 (2-3), 211–247.

[39] Sieg, A., Mobasher, B., Burke, R., 2007. Web search personalization with ontological user profiles. In: 16th ACM Conference on Information and Knowledge Management. CIKM '07. ACM, pp. 525–534.

[40] Sieg, A., Mobasher, B., Burke, R., 2010. Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In: 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems. HetRec '10. ACM, pp. 39–46.

[41] Sosnovsky, S., Dicheva, D., 2010. Ontological technologies for user modelling. International Journal of Metadata Semantic Ontologies 5 (1), 32–71.

[42] Swinehart, D., 1962. The deer-lambert law. Journal of Chemical Education 39 (7), 333.

[43] Welty, C. A., Guarino, N., 2001. Supporting ontological analysis of taxonomic relationships. Data Knowledge Engeneering 39 (1), 51–74.

[44] Xue, G.-R., Zeng, H.-J., Chen, Z., Yu, Y., Ma, W.-Y., Xi, W., Fan, W., 2004. Optimizing web search using web click-through data. In: 13th ACM International Conference on Information and knowledge management. CIKM '04. ACM, pp. 118–126.

[45] Zhang, F., Song, Z., Zhang, H., 2006. Web service based architecture and ontology based user model for cross-system personalization. In: IEEE/WIC/ACM International Conference on Web Intelligence, WI '06. IEEE Computer Society, pp. 849–852.